REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
14-06-2017	Final Report	1-Sep-2013 - 31-Aug-2016

4. TITLE AND SUBTITLE
Final Report: Searching Information Sources in Networks

5a. CONTRACT NUMBER
W911NF-13-1-0279

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Lei Ying

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Arizona State University
ORSPA
P.O. Box 876011
Tempe, AZ 85287 -6011

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
63873-NS.13

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
During the course of this project, we made significant progresses in multiple directions of the information detection problem. These progresses include: (1) The development of the first algorithm that is asymptotic optimal for the Erdos-Renyi (ER) random graph when the infection time is less than 2/3t_u. The algorithm is called the Short-Fat-Tree (SFT) algorithm and is the first algorithm and the first theoretical result on information source detection on non-tree networks; (2) The development of information source localization algorithms to detect multiple information sources. The algorithms have provable performance guarantees and outperform existing algorithms in

15. SUBJECT TERMS
source localization, large-scale graphs, partial information, general diffusion models

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Lei Ying
UU	UU	UU	UU		19b. TELEPHONE NUMBER 480-965-7003

RPPR

as of 03-Oct-2017

Agency Code:

Proposal Number: **Agreement Number:**

Organization:
Address: , ,
Country:
DUNS Number: EIN:
Report Date: Date Received:
for Period Beginning and Ending
Title:
Begin Performance Period: **End Performance Period:**
Report Term: -
Submitted By: Email:
Phone:

Distribution Statement: -

STEM Degrees: **STEM Participants:**

Major Goals:

Accomplishments:

Training Opportunities:

Results Dissemination:

Plans Next Period:

Honors and Awards:

Protocol Activity Status:

Technology Transfer:

Scientific Progress and Accomplishments

1 Summary

During the course of this project, we made significant progresses in multiple directions of the information detection problem. These progresses include: (1) The development of the first algorithm that is asymptotic optimal for the Erdos-Renyi (ER) random graph when the infection time is less than $2/3\ t_u$. The algorithm is called the Short-Fat-Tree (SFT) algorithm and is the first algorithm and the first theoretical result on information source detection on non-tree networks; (2) The development of information source localization algorithms to detect multiple information sources. The algorithms have provable performance guarantees and outperform existing algorithms in the literature; (3) The extension of the sample-path-based estimator to the heterogeneous SIR model with sparse observations, where we showed that the Jordan infection center remains to be the sample-path-based estimator so is a robust source estimator under heterogeneous diffusion models and partial information; (4) The development of the sample-path-based estimator with partial observations including partial timestamps. We developed two ranking algorithms (tree-based ranking and cost-based ranking) that rank infected nodes according to their likelihood of being the source when partial infection timestamps are available; and (5) The development of an algorithm based on the sample-path-based approach and a correlation network to identify the root cause of cascading failures in power networks.

2 Information Source Detection in General Networks

The source localization problem has gained a lot of attention in the last few years. A number of source localization algorithms have been developed under different diffusion models. However, despite significant efforts and successes, theoretical guarantees have been established only for tree networks due to the complexity of the problem. In our recent work [1], we developed a new source localization algorithm, called the Short-Fat Tree (SFT) algorithm for general networks. Consider the Erdos-Renyi (ER) random graph and assume the average node degree $\mu = \Omega(\log n)$ to guarantee the connectivity of the network, where n is the number of nodes. We established the following fundamental limits of SFT on the ER random graph.

(1) When the infection duration $< \frac{2}{3}\frac{\log n}{\log \mu}$, SFT identifies the source with probability one (w.p.1) asymptotically (as network size increases).

(2) When the infection duration $\geq \frac{\log n}{\log \mu} + 2$, the probability of identifying the source approaches zero asymptotically under *any* source localization algorithm, i.e., it is *impossible* to identify the source with a non-zero probability.

(3) When the infection duration $< \frac{\log n}{\log \mu}$, asymptotically, at least $1 - \delta$ fraction of the nodes on the BFS-tree starting from the source are leaf-nodes, where $\delta > 3\sqrt{\frac{\log n}{\mu}}$. Note that this result does not provide a guarantee on the probability of correctly localizing the source, but states that

the BFS-tree starting from the true source is a "fat" tree, which justifies the SFT algorithm, and is confirmed by the simulation results.

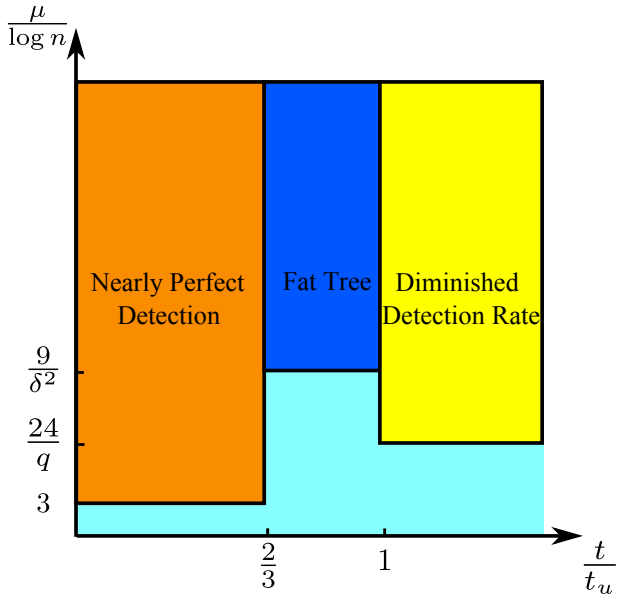


Figure 1: Summary of the main results of source localization in [1]. This figure summarizes the key results in terms of t, the infection time, and μ, the average node degree. In the figure, $t_u = \frac{\log n}{\log \mu}$.

The results are summarized in Figure 1. We remark that results (1) and (3) are highly nontrivial because a subgraph of the ER random graph is a tree with high probability *only when the diameter of the subgraph is* $\frac{\log n}{2 \log \mu}$. Results (1) and (3) deal with subgraphs that are not trees. *To the best of our knowledge, these are the first theoretical results of source localization on general (non-tree) networks*. A comparison of detection rates with other source localization algorithms on an ER random graph with $5,000$ nodes and under the IC diffusion model is shown in Figure 2. We can see that SFT has a detection rate ≥ 0.9 when the number of infected nodes is $\leq 1,200$, ECCE [2] achieves a similar detection rate when the number of infected nodes is ≤ 600, i.e., only a half of that under SFT, and the detection rates of RUM [3] and NETSLEUTH [4] are always less than 0.9.

The key to accurately locating the diffusion source is to identify characteristics of infection sub-networks that are unique "signatures" of the source. The novelty of [1] is at the use of *frontier nodes* for source localization, where frontier nodes are the set of the infected nodes that are furthest away from the source. For example, in Figure 3 where node a is the source, nodes c and d are frontier nodes. They are two hops away from the source. All remaining infected nodes are one hop away from the source. In [1], we identified two properties of frontier nodes.

Distance-signature: If the diffusion has not died out when the observation was taken (at time slot t), then the frontier nodes are the ones infected at time t (but not all infected nodes infected at time slot t are frontier nodes). With a high probability, frontier nodes are t hops away from the source. Define $\mathcal{B}(v,t)$ to be the set of infected nodes that are t-hop away from node v. Assume node v is the diffusion source. Then $\mathcal{B}(v,t) \cap \mathcal{I}$ is the set of frontier nodes, where $\mathcal{I}$ is the set of infected nodes. Consider two nodes v and w and define $\mathcal{D}(v,w,t) = \mathcal{B}(v,t) \backslash \mathcal{B}(w, \leq t)$, which is the set of nodes that are t hops away from node v, but more than t hops away from node w. Then as long as $\mathcal{D}(v,w,t) \cap \mathcal{I}$ is a nonempty set, node w cannot reach all frontier nodes within t hops. We proved in [1] that $\mathcal{D}(v,w,t) \cap \mathcal{I} \neq \emptyset$ with a high probability by analyzing the distribution of the frontier nodes in the ER random graph. Let $d(v,\mathcal{I}) = \max_{u \in \mathcal{I}} d(v,u)$, where $d(v,u)$ is the minimum number of hops to reach node u from v. Then $d(v,\mathcal{I})$is a signature of the source such that $d(v,\mathcal{I}) < d(u,\mathcal{I})$ for any other u when $t \leq \frac{2}{3} \frac{\log n}{\log \mu}$ and $d(v,\mathcal{I}) \leq d(u,\mathcal{I})$ when $t \leq \frac{\log n}{\log \mu}$.

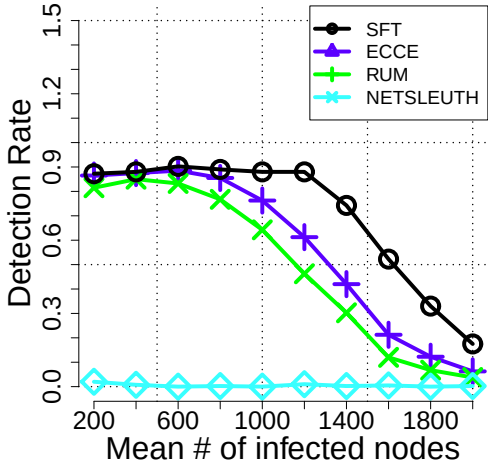


Figure 2: Detection rates of different source localization algorithms

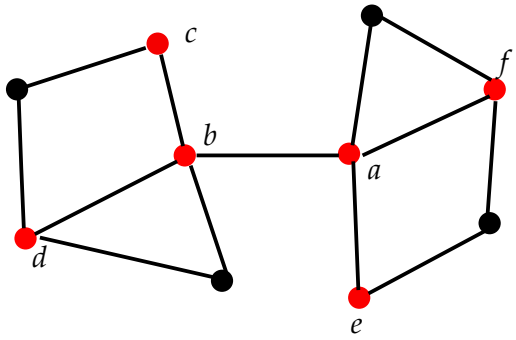


Figure 3: Example that illustrates the frontier nodes

Size-signature: It has also been identified in [1] that among infected nodes with the smallest distance-signature, the node has more frontier nodes (i.e., larger $|\mathcal{B}(v,t) \cap \mathcal{I}|$) is more likely to the source. Intuitively, "fat" diffusion frontier often associates with a "symmetric and smooth" diffusion trace, which is likely to occur.

3 Detecting Multiple Information Sources under Heterogeneous Diffusion Models

In [5], we studied the information source detection problem in a setting that generalizes the existing work in several important directions.

- *Multiple sources versus single source:* We assumed that the information diffusion can be originated from multiple nodes simultaneously, instead of from a single source. When the infection duration is sufficiently short, the infected subnetworks from different sources are disconnected components. In such cases, the single-source localization algorithms can be applied to each of the infected subnetwork. We removed this assumption in [5], and considered the scenario where the infected subnetworks may overlap with each other, so the single-source localization algorithms cannot be directly applied.

- *A partial snapshot versus a complete snapshot:* We assumed a partial snapshot in which each node reports its state with some probability, which is in contrast to a complete snapshot assumed in the literature where all nodes' states are observed. Because of a partial snapshot, the sources may not report their states and be observed as infected nodes; and the observed infected nodes may not form a connected component. Both increase the uncertainty and complexity of the problem. In fact, it turns out to be critical to have a candidate selection algorithm to select source candidates from unobserved nodes but only use observed infected nodes in computing the infection eccentricity. The selection step yields $27\times$ reduction on the computing time in our simulations while guaranteeing the same the detection rate, and yields $600\times$ reduction on the computing time with a slight reduction of the detection rate.

- *Heterogeneous diffusion versus homogeneous diffusion:* Our algorithm applies to the heterogeneous SIR diffusion model where links have different infection probabilities and nodes have different recovery probabilities. The asymptotic guarantees on the detection rate hold for the heterogeneous SIR model.

In [5], we developed a novel algorithm for locating multiple sources for such a general model and proved theoretical guarantees on the detection rate for non-tree networks. The main results of the paper are summarized below.

(1) We introduced the concept of Jordan cover, which is an extension of Jordan center. Loosely speaking, a Jordan cover with size m is a set of m nodes that can reach all *observed* infected nodes with the minimum hop-distance. We proposed Optimal-Jordan-Cover (OJC), which consists of two steps: OJC first selects a subset of nodes as the set of the candidates of the diffusion sources; and then it finds a Jordan cover in the subgraph induced by the candidate nodes and the observed infected nodes. We emphasize that only the hop-distance to the observed infected nodes is considered in computing a Jordan cover.

(2) We analyzed the performance of OJC on the ER random graph, and established the following performance guarantees.

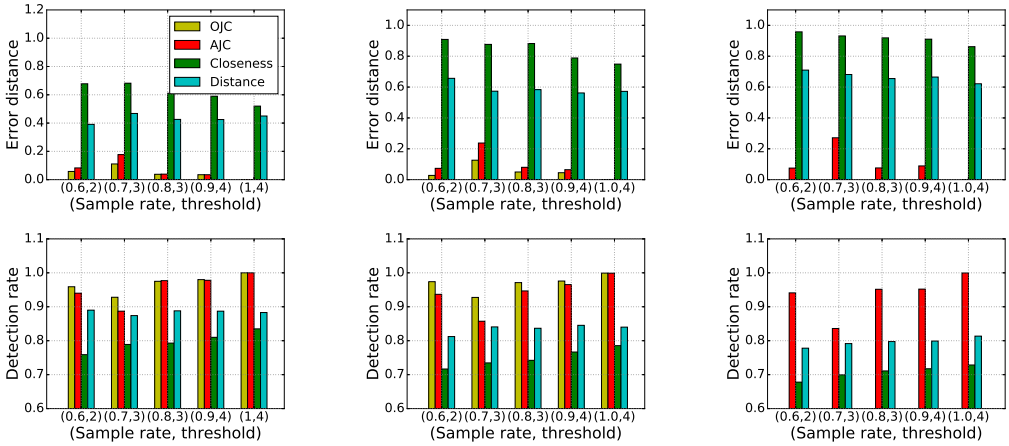
   (i) When the infection duration is shorter than $\frac{2}{3}\frac{\log n}{\mu}$, where μ is the average node degree and n is the number of nodes in the network, OJC identifies the sources with probability one asymptotically as n increases.

   (ii) When the infection duration is at least $\left\lceil \frac{\log n}{\log \mu + \log q} \right\rceil + 2$ where q is the minimum infection probability, *under any source location algorithm,* the detection rate diminishes to zero as n increases under the Susceptible-Infected (SI) and Independent-Cascade (IC) models, which are special cases of the SIR model.

(3) The computational complexity of OJC is polynomial in n, but exponential in m. We further proposed a heuristic based on the K-Means for approximating the Jordan cover, named Approximate-Jordan-Cover (AJC). Assuming a constant number of iterations when using the K-Means, the computational complexity of AJC is $O(nE)$, where E is the number of edges. Our simulations on random graphs and real networks demonstrate that both AJC and OJC significantly outperform other heuristic algorithms.

In Figure 4 and 5 show the performance of our algorithms, OJC and AJC, on both the power grid network [6] and ER random graph (size: 5000, wiring probability: 0.002) and compared them with two other heuristics — Distance-Centroid-Based K-Means (DC) and Closeness-Centroid-Based K-Means (CC). The $x-$axis of the figures represents the combinations of sample rate and threshold. On the ER random graph, we increased the threshold as the sample rate increased to control the running time. For the power-grid network, since the average node degree is only 2, we set threshold equal to 2 for experiments for all sample rates. As we can see from the figures that when fixing the threshold, the performance of all algorithms (in terms of both error distance and detection rate) improves as the sample rate increases because we had more information about the diffusion. From Figure 4 and 5, we can also see that AJC outperforms DC and CC, and has similar performance with OJC. Note that with four sources, OJC became very slow on both the ER random graph and the power grid network because its complexity increases exponentially in the number of sources. So for the cases with four sources, we only simulated AJC.

Besides [5], in an earlier work [7, 8], we also extended the sample-path-based estimator developed in [9, 10] to multiple information sources under the SIR model. The algorithm includes both clustering and source localization. The clustering and localization algorithm first iteratively selects S infected nodes in a greedy fashion to maximize the pairwise distances of these S nodes. These S

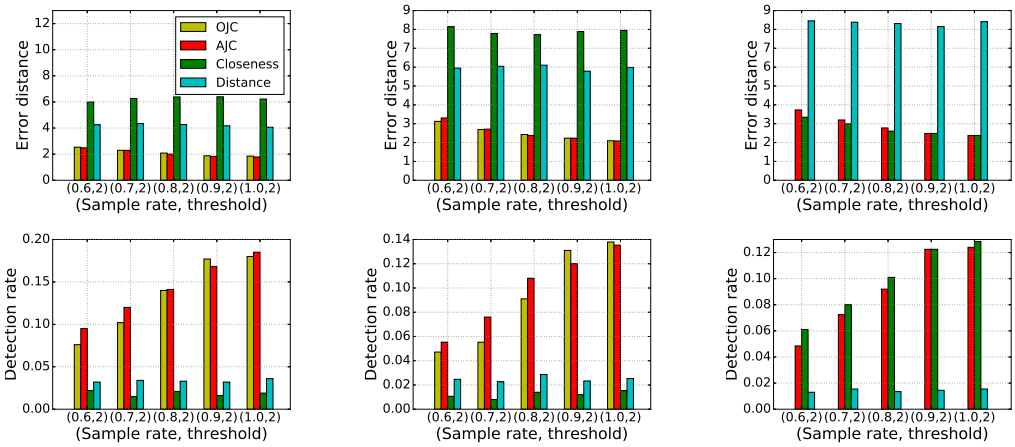


(a) Source number: 2, infection size: $100 \sim 300$.

(b) Source number: 3, infection size: $200 \sim 400$.

(c) Source number: 4, infection size: $300 \sim 500$.

Figure 4: The Performance of OJC, AJC, CC and DC on the ER random graph with different sample rates and threshold values



(a) Source number: 2, infection size: $100 \sim 300$.

(b) Source number: 3, infection size: $200 \sim 400$.

(c) Source number: 4, infection size: $300 \sim 500$.

Figure 5: The Performance of OJC, AJC, CC and DC on the power grid network with different sample rates and threshold values

infected nodes are likely to be associated with different sources, and are likely to be the leaf nodes of the corresponding information spreading trees. Then, the set of infected nodes are clustered into S sets according to their distances to the selected S nodes. The purpose is to cluster the infected nodes according to their associated sources. In the final step, the reverse infection algorithm proposed in [9] is used to identify the Jordan infection center within each cluster and these Jordan infection centers are claimed to be the information sources. The algorithm above assumes the number of sources is known. In [7], we also developed an algorithm for the case in which the number of sources is unknown.

4 Information Source Detection with Sparse Observations and Heterogeneous SIR Model

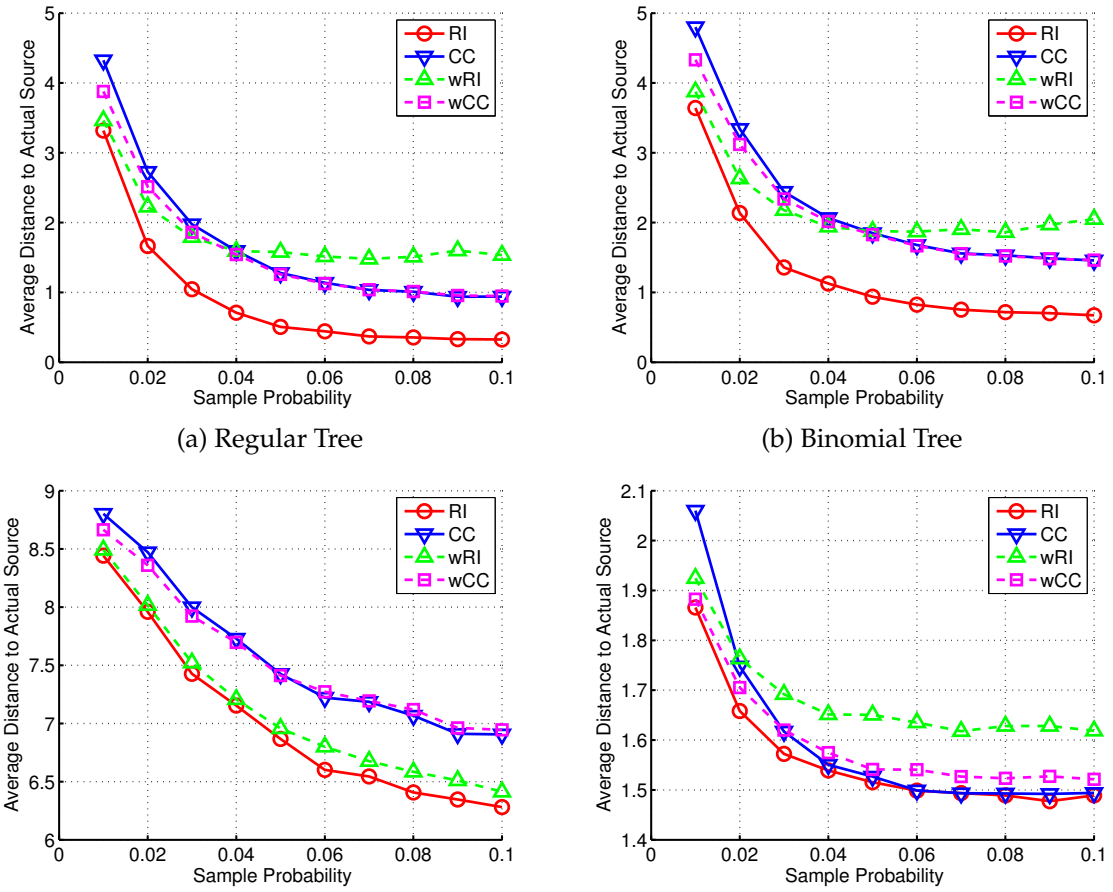
In [2, 11], we studied the problem of locating the source of information that spreads in a network by using sparse observations. In [9, 10], we assume a complete snapshot of the network is given. Today's online social network may have hundreds of millions of nodes, so a complete snapshot is expensive, if not impossible, to obtain. Furthermore, most previous works on information source detection assume homogeneous infection across links and homogeneous recovery across nodes, but in reality, most networks are heterogeneous. For example, people close to each other are more likely to share rumors and epidemics are more infectious in the regions with poor medical care systems. Therefore, it is important to take sparse observations and network heterogeneity into account when locating information sources. In this work, we assumed that the information spreads in the network following a heterogeneous SIR model and assume only a small subset of infected nodes are reported to us. The goal is to identify the information source in a heterogeneous network by using sparse observations.

We used the sample path based approach developed in [9] for locating the information source with sparse observations. *Surprisingly, we found that the sample path based estimator is robust to network heterogeneity and the number of observed infected nodes.* In particular, our results show that even under a heterogeneous SIR model and with sparse observations, the sample path based estimator remains to be a Jordan infection center in infinite trees, where the Jordan infection centers with a partial observation are the nodes that minimize the maximum distance to observed infected nodes. We further showed that in an infinite tree, the distance between a Jordan infection center and the actual source can be bounded by a value independent of the size of infected subnetwork with a high probability, where the infected subnetwork is the subnetwork that consists of nodes are either infected or recovered and is a connected component.

We remark that the locations of the Jordan centers only depend on the network topology and are independent of the infection and recovery probabilities, so the sample path based estimators (or the Jordan infection centers) are also robust to the information diffusion model, which makes it very appealing in practice since the accurate knowledge of the SIR parameters can be difficult to measure in reality.

In [2, 11], we conducted extensive simulations to evaluate the performance of the Jordan infection center for the heterogeneous SIR model on different networks including tree networks and real world networks. In the simulations, we compared the performance of our algorithm with the following three algorithms.

- *Closeness centrality algorithm (CC)*: The closeness centrality algorithm selects the node with the maximum infection closeness as the information source.
- *Weighted reverse infection algorithm (wRI)*: The weighted reverse infection algorithm selects the node with the minimum weighted infection eccentricity as the information source where the weighted infection eccentricity is similar to the infection eccentricity except that the length of



(a) Regular Tree

(b) Binomial Tree

(c) The Power Grid Network

(d) The Internet Autonomous Systems Network

Figure 6: The Performance of RI, CC, wRI and wCC on Different Graphs

a path is defined to be the sum of the link weights instead of the number of hops, and the link weight is the average time it takes to spread the information over the link, i.e., $\lfloor 1/q_e \rfloor$ on edge e, where q_e is the infection probability of link e.

- *Weighted closeness centrality algorithm (wCC)*: The weighted closeness centrality algorithm selects the node with the maximum weighted infection closeness as the information source.

We first evaluated the performance on tree networks. A g-regular tree is a tree where each node has g neighbors. We set the degree $g = 5$ in our simulations. We varied the sample probability σ from 0.01 to 0.1. The simulation results are summarized in Figure 6a, which shows the average distance between the estimator and the actual information source versus the sampling probability. When the sample probability increases, the performance of all algorithms improve. When the sample probability is larger than 6%, the average distance becomes stable which means a small number of infected nodes is enough to obtain a good estimator. We also notice that the average distance of our algorithm based on the Jordan infection center, named RI, is smaller than all other algorithms and is less than one hop when $\sigma \geq 0.04$. wRI has a similar performance with RI when the sample probability is small (=0.01) but becomes much worse when the sample probability increases.

We further evaluated the performance of RI and other algorithms on binomial trees $T(\xi, \beta)$ where the number of children of each node follows a binomial distribution such that ξ is the number of trials and β is the success probability of each trial. In the simulations, we selected $\xi = 10$

and $\beta = 0.4$. Again, we varied σ from 0.01 to 0.1. The results are shown in Figure 6b. Similar to the regular trees, the performance of RI dominates CC, wRI and wCC, and the difference in terms of the average number of hops is approximately one when $\sigma \geq 0.03$.

We also conducted experiments on two real world networks: the Internet Autonomous Systems network (IAS) [12] and the power grid network (PG) [13]. The results for the power network are shown in Figure 6c. In the power grid network, we can see that RI and wRI have similar performance, and both outperform CC and wCC by at least one hop when $\sigma \geq 0.04$. The results for the IAS network are shown in Figure 6d. wRI and wCC always perform worse than RI. Although RI and CC have similar performance when the sample probability is large, RI outperforms CC when $\sigma \leq 0.03$.

We further compared the performance of RI and DMP. The dynamic message passing algorithm (DMP) algorithm was proposed in a recent paper [14] for a general SIR model with complete or partial observations. However, the algorithm needs the complete information of infection and recovery probabilities, and the complexity of the algorithm is very high under partial observations. We compared the performance of the two algorithms on the power grid network and fixed the sample probability to be 10%. We tested the speed of RI and DMP on a machine with 1.8 GB memory, 4 cores 2.4 GHz Intel i5 CPU and Ubuntu 12.10. The algorithms are implemented in Python 2.7. On average, it took RI 0.57 seconds to locate the estimator for one snapshot and took DMP 229.12 seconds. So RI is much faster than DMP.

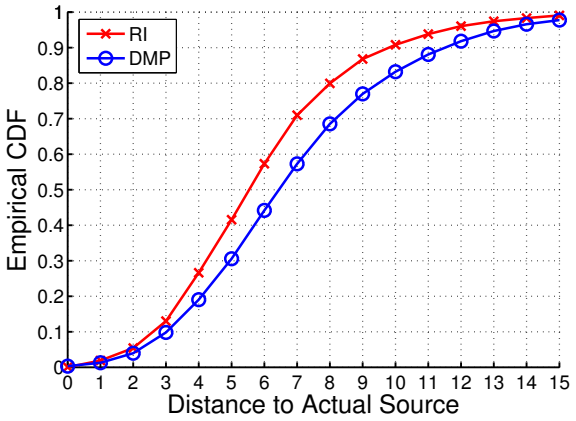


Figure 7: The CDF of RI and DMP on the Power Grid Network

Figure 7 shows the CDF of the distance from the estimator to the actual source under DMP and RI. We can see that RI dominates DMP, in particular, 71% of the estimators under RI are no more than 7 hops from the actual source comparing to 57% under DMP. Therefore, RI outperforms DMP in terms of both speed and accuracy. We remark that we did not compare the performance of RI and DMP on the Internet Autonomous System (IAS) network because the complexity of running DMP on a large size network like the IAS network is prohibitively high.

5 Information Source Detection with Partial Timestamps

A major challenge of locating the information sources(s) is the lack of complete timestamp information, which prevents us from reconstructing the spreading sequence to trace back the source. But on the other hand, even partial timestamps, which are available in many practical scenarios, provide important insights about the location of the source. In [15], we developed source localization algorithms that utilize partial timestamp information. The main accomplishments are summarized below.

(1) We formulated the source localization problem as a ranking problem on graphs, where infected nodes are ranked according to their likelihood of being the source. Define a *spreading tree* to include (i) a directed tree with all infected nodes and (ii) the complete timestamps of information spreading. Given a spreading tree rooted at node v, denoted by $\mathcal{P}_v$, we defined a quadratic cost $C(\mathcal{P}_v)$ depending on the structure of the tree and the timestamps. The cost of node v is then defined to be

$$C(v) = \min_{\mathcal{P}_v} C(\mathcal{P}_v), \tag{1}$$

i.e., the minimum cost among all spreading trees rooted at node v. Based on the costs and spreading trees, we propose two ranking methods:

(i) rank the infected nodes in an ascendent order according to $C(v)$, called *cost-based ranking (CR)*, and

(ii) find the minimum cost spreading tree, i.e.,

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} C(\mathcal{P}),$$

and rank the infected nodes according to their timestamps on the minimum cost spreading tree, called *tree-based ranking (TR)*.

(2) The computational complexity of $C(v)$ is very high due to the large number of possible spreading trees. We proved that problem (1) is NP-hard by connecting it to the longest-path problem.

(3) We proposed a greedy algorithm, named Earliest Infection First (EIF), to construct a spreading tree to approximate the minimum cost spreading tree for a given root node v, denoted by $\tilde{\mathcal{P}}_v$. The greedy algorithm is designed based on the minimum cost solution for line networks. EIF first sorts the infected nodes with observed timestamps in an ascendent order of the timestamps, and then iteratively attaches these nodes using a modified breadth-first search algorithm. In CR, the infected nodes are then ranked based on $C(\tilde{\mathcal{P}}_v)$; and in TR, the nodes are ranked based on the complete timestamps of the spreading tree $\tilde{\mathcal{P}}^*$ such that

$$\tilde{\mathcal{P}}^* = \arg\min C(\tilde{\mathcal{P}}_v).$$

In [15], we evaluated the performance of TR and CR (based on EIF) using both synthetic data and real-world data. To evaluate the accuracy of the ranking, we use the probability that the source is ranked among top $\gamma\%$ of the infected nodes, named $\gamma\%$-*accuracy*, as the performance metric. The main results and observations are summarized below.

Comparison with Existing Algorithms

We first tested the algorithms using synthetic data on two real-world networks: the Internet Autonomous Systems network (IAS) and the power grid network (PG), and compared with the following four existing source localization algorithms.

- Rumor centrality (RUM): Rumor centrality was proposed in [3], and is the the maximum likelihood estimator on trees under the SI model. RUM ranks the infected nodes in an ascendent order according to nodes' rumor centrality.

- Infection eccentricity (ECCE): The infection eccentricity of a node is the maximum distance from the node to any infected node in the graph, where the distance is defined to be the length of the shortest path. The node with the smallest infection eccentricity, named Jordan infection center, is the optimal sample-path-based estimator on tree networks under the SIR model [9]. ECCE ranks the infected nodes in a descendent order according to infection eccentricity.

- NETSLEUTH: NETSLEUTH was proposed in [4]. The algorithm constructs a submatrix of the infected nodes based on the graph Laplacian of the network and then ranks the infected nodes according to the eigenvector corresponding to the largest eigenvalue of the submatrix.
- Gaussian heuristic (GAU): Gaussian heuristic is an algorithm proposed in [16], which utilizes partial timestamp information. The algorithm is similar to CR in spirit, but uses the breadth-first search tree as the spreading tree for each infected node.

In the four algorithms above, RUM, ECCE, and NETSLEUTH only use topological information of the network, and do not exploit the timestamp information. GAU utilizes partial timestamp information.

In this set of experiments, we assumed the infection time of each infection follows a truncated Gaussian distribution with $\mu = 100$ and $\sigma = 100$. We selected 50% infected nodes (100 nodes) and revealed their infection time. The source node was always excluded from these 100 nodes so that the infection time of the source node was always unknown. We repeated the simulation 500 times to compute the average $\gamma\%$-accuracy.

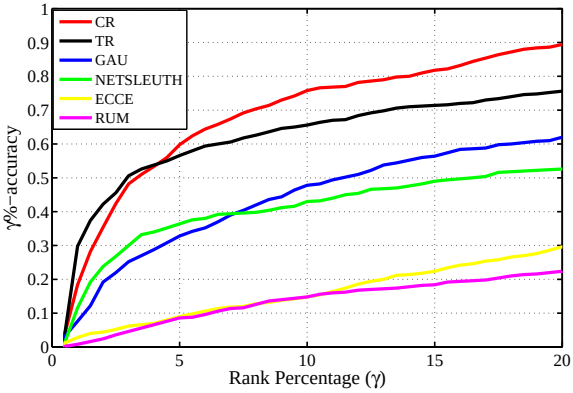


Figure 8: Comparison with Existing Algorithms in the IAS network with 50% timestamps

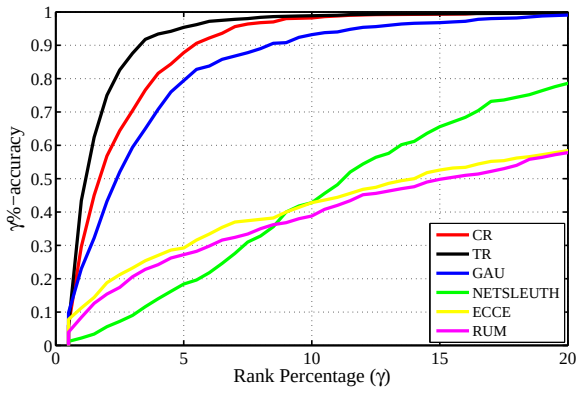


Figure 9: Comparison with Existing Algorithms in the PG network with 50% timestamps

The results on the IAS and PG networks are presented in Figures 8 and 9, respectively. Recall that RUM, ECCE and NETLEUTH only use topological information.

- **Observation 1:** In both networks, CR and TR perform much better than the other algorithms. In particular, in the IAS network, the 10%-accuracy of CR is 0.76 while 10%-accuracy of GAU and NETSLEUTH is 0.48 and 0.43, respectively.In the PG network, the 10%-accuracy of TR is 0.99 while that of GAU and NETSLEUTH is 0.93 and 0.43, respectively.
- **Observation 2:** Most algorithms, except NETSLEUTH, have higher $\gamma\%$-accuracy in the PG network than in the IAS network. We conjecture that it is because the IAS network has a small diameter and contains hub nodes while the PG network is more tree-like.
- **Observation 3:** NETSLEUTH dominates ECCE and RUM in the IAS network, but performs worse than ECCE and RUM in the PG network when $\gamma \leq 10$. Furthermore, while all other algorithms have higher γ-accuracy in IAS than in PG, NETSLEUTH has lower γ-accuracy in IAS than in PG when $\gamma < 10$. A similar phenomenon will be observed in a later simulation as well.
- **Observation 4:** CR performs better in the IAS network when $\gamma \geq 5$ while TR performs better in the PG network.

The Impact of Timestamp Distribution

In the previous set of simulations, the revealed timestamps were uniformly chosen from all timestamps except the timestamp of the source, which was always excluded. We call this *unbiased distribution.* In this set of experiments, we studied the impact of the distribution of the timestamps. We compared the unbiased distribution with a distribution under which nodes with larger infection time are selected with higher probability. In particular, we selected nodes iteratively. Let $\mathcal{N}^k$ denote the set of remaining infected nodes after selecting k nodes, then the probability that node i is selected in the next step is

$$p_i^{(k)} = \frac{t_i - t_s}{\sum_{j \in \mathcal{N}^k}(t_j - t_s)},$$

where t_s is the infection time of the source. We call this *time biased distribution*.

In addition, we evaluated the performance of our algorithms and GAU with different sizes of observed timestamps. The results for IAS and PG are shown in Figure 10 and Figure 11, respectively. Note that the performance of RUM, ECCE and NETSLEUTH are independent of timestamp distribution and size, so we did not include these algorithms in the figures. From the two figures, we have the following observations:

- **Observation 5:** We varied the size of observed timestamps from 10% to 90%. As we expected, the $\gamma\%$-accuracy increases as the size increases under both CR and TR. Interestingly, in the IAS network, the 10%-accuracy of GAU only has minor improvement when the timestamp size increases. We conjecture this is because in small world networks such as the IAS network, the spreading tree is very different from the breadth-first search tree rooted at the source. Since GAU always uses the breadth-first search trees regardless of the size of timestamps, more timestamps do not result in a more accurate spreading tree. The spreading tree constructed by EIF, on the other hand, depends on the size of timestamps and is more accurate as the size of timestamps increases.

- **Observation 6:** In both networks, the time-biased distribution results in 5% to 15% reduction of the $\gamma\%$-accuracy. This shows that earlier timestamps provide more valuable information for locating the source. However, the trends and relative performance of the three algorithms are similar to those in the unbiased case.

- **Observation 7:** CR performs better in the IAS network when the timestamp size is larger than 40%; and TR performs better in the PG network.

- **Observation 8:** The $\gamma\%$-accuracy is much higher in the PG network than that in the IAS network under both the unbiased distribution and time-biased distribution. For example, with the time-biased distribution and 20% of timestamps, the 10%-accuracy of TR is 0.71 in PG and is only 0.41 in IAS. This again confirms that the source localization problem is more difficult in networks with small diameters and hub nodes.

The Impact of the Diffusion Model

In all previous experiments, we used the truncated Gaussian model for information diffusion. We now study the robustness of CR and TR to the diffusion models. We conducted the experiments using the SpikeM model [17], which matches the patterns of real-world information diffusion well. The results are shown in Figures 12 and 13.

- **Observation 9:** SpikeM is a time slotted diffusion model, so is very different from the truncated Gaussian model. However, as shown in Figures 12 and 13, the $\gamma\%$-accuracy does not

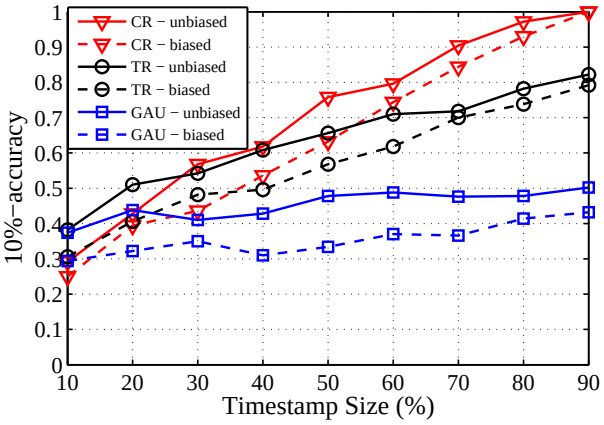


Figure 10: The Impacts of the Distribution and Size of Timestamps in the IAS Network

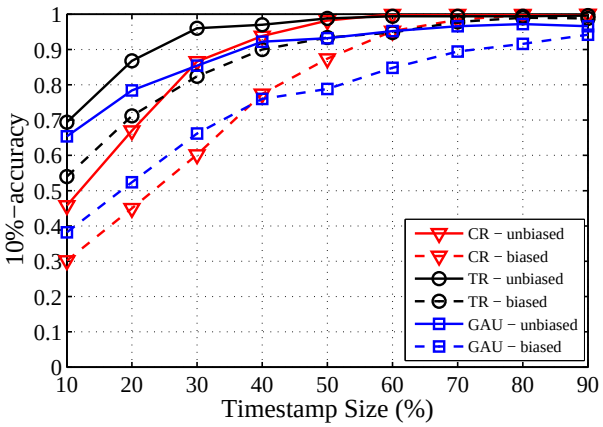


Figure 11: The Impacts of the Distribution and Size of Timestamps in the PG Network

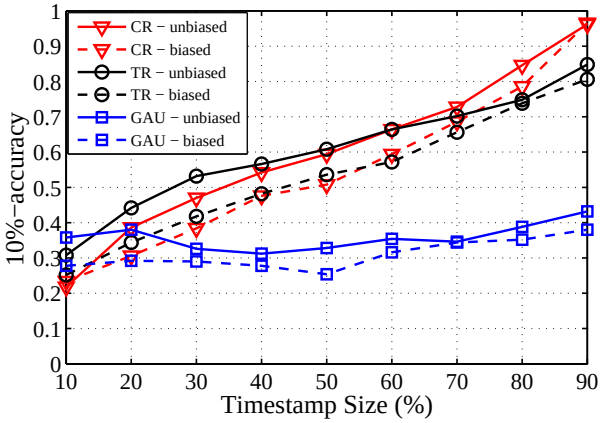


Figure 12: The Performance of CR, TR and GAU in the IAS Network under the SpikeM Model

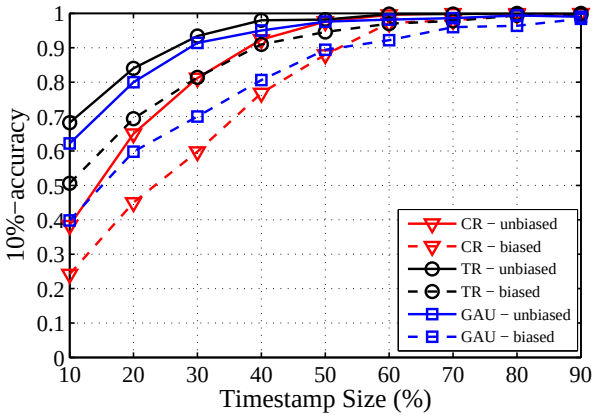


Figure 13: The Performance of CR, TR and GAU in the PG Network under the SpikeM Model

change significantly compared with the truncated Gaussian model. Therefore, CR and TR are robust to the diffusion model. We remark that both algorithms do not require any knowledge of the underlying diffusion model.

The Impact of Network Topology

In the previous simulations, we have observed that locating the source in the PG network is easier than in the IAS network. We conjecture that it is because the IAS network is a small-world network while the PG network is more tree-like. To verify this conjecture, we removed edges from the IAS network to observe the change of $\gamma\%$-accuracy as the number of removed edges increases. For each removed edge, we randomly picked one edge and removed it if the network remains to be connected after the edge is removed. The results are shown in Figure 14.

- **Observation 10:** After removing 11,000 edges, the ratio of the number of edges to the number of nodes is $11,002/10,670 = 1.03$, so the network is tree-like. As showed in Figure 14, the 5%-accuracy of all algorithms, except NETSLEUTH, improves as the number of the removed edges increases, which confirms our conjecture. The 5%-accuracy of NETSLEUTH starts to decrease when the number of removed edges is more than $6,000$. This is consistent with the

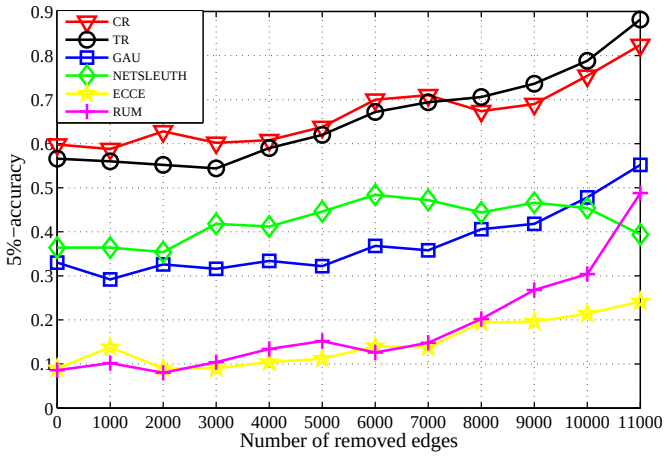


Figure 14: The $\gamma\%$-Accuracy as the Number of Removed Edges Increases

observation we had in Figures 8 and 9, in which the 5% accuracy of NETSLUETH in PG is worse than that in IAS.

Weibo Data Evaluation

In this section, we evaluated the performance of our algorithms with real-world network and real-world information spreading. The dataset is the Sina Weibo[1] data, provided by the WISE 2012 challenge[2]. Sina Weibo is the Chinese version of Twitter, and the dataset includes a friendship graph and a set of tweets.

The friendship graph is a directed graph with 265,580,802 edges and 58,655,849 nodes. The tweet dataset includes 369,797,719 tweets. Each tweet includes the user ID and post time of the tweet. If the tweet is a retweet of some tweet, it includes the tweet ID of the original tweet, the user who post the original tweet, the post time of the original tweet, and the retweet path of the tweet which is a sequence of user IDs. For example, the retweet path $a \to b \to c$ means that user b retweeted user a's tweet, and user c retweeted user b's.

We selected the tweets with more than 1,500 retweets. For each tweet, all users who retweet the tweet are viewed as infected nodes and we extracted the subnetwork induced by these users. We also added those edges on the retweet paths to the subnetwork if they are not present in the friendship graph, by treating them as missing edges in the friendship network. The user who posts the original tweet is regarded as the source. If there does not exist a path from the source to an infected node along which the post time is increasing, the node was removed from the subnetwork.

Note that in some cases, the source can be easily located using a naive algorithm, e.g., the network is star or only the source can reach all other infected nodes. To avoid these cases, we further selected the tweets that satisfy the following conditions:

- The number of infected nodes is at least 100.
- The diameter of the undirected version of the subnetwork network is at least 7.
- There exist at least 50 nodes who can reach all other infected nodes in the network.
- At least 30% of nodes have timestamps. This is to make sure we have enough timestamps for evaluating CR and TR.

[1] http://www.weibo.com/

[2] http://www.wise2012.cs.ucy.ac.cy/challenge.html

After removing tweets that do not satisfy the above conditions, we have 347 tweets with at least 30% observed timestamps. The $\gamma\%$-accuracy is summarized in Figure 15, where we include the results with 10% of timestamps and 30% of timestamps. The observed timestamps are uniformly selected from the available timestamps and the source node is excluded. Note that for most of the tweets, we only have partial timestamps, so we believe further biased selection is not necessary. RUM and ECCE are not included in the figure since the performance is dominated by NETSLEUTH.

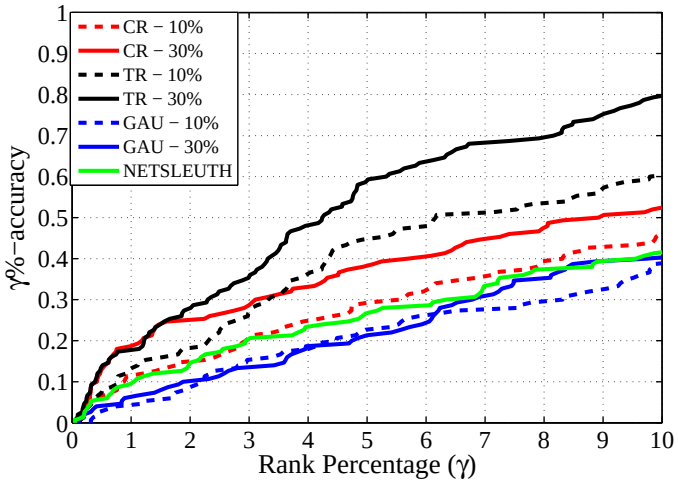
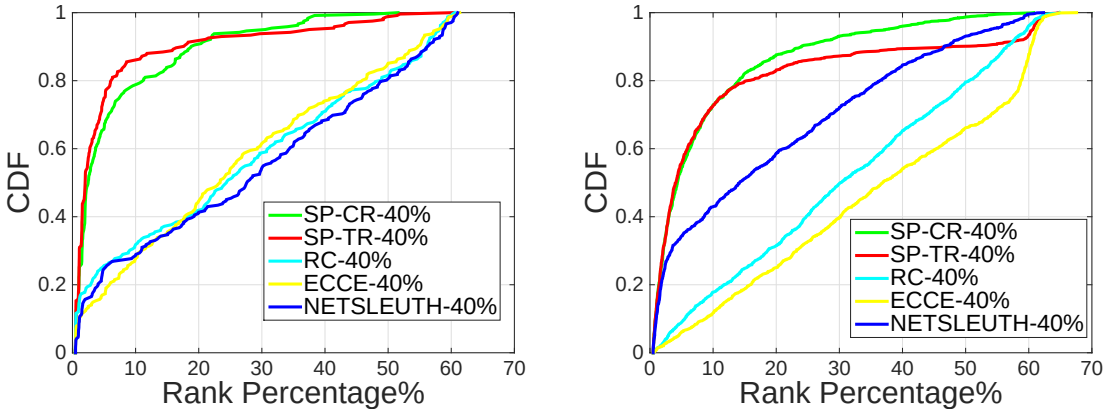


Figure 15: Performance on Weibo Data

- **Observation 10:** The figure shows that CR and TR dominates GAU with both 10% and 30% of timestamps. In particular, TR performs very well and dominates all other algorithms with a large margin. The 10%-accuracy of TR with 30% timestamps is around 0.81 while that of CR is 0.53 and that of NETSLEUTH is only 0.39.

6 Root Cause of Cascading Failures in Power Networks

The power grid is a critical infrastructure of our society. The failure of the power grid network will have catastrophic impacts on water supplies, transportation networks, communications networks, and almost every other aspect of our daily life. For post-cascade fault diagnostics, important questions shall be answered including: How can we locate the source of fault on the power network? Which kind of information can we utilize to locate the initially malfunctioning part that causes the cascading failures? The answers to these questions are crucial for recovering the power network from cascading failures and even stopping the cascade. In [18], we studied the problem of locating the root cause of cascading failures in power networks, and related the problem to source localization in networks. Note that the root-cause localization problem is different from the information source detection problem. In social networks, the information generally propagates through topological connections while the cascading failures in power networks are the result of electrical interactions restricted by Kirchhoff's and Ohm's laws. In addition, it has been pointed out in the literature that topological models may result in misleading conclusions in power networks. In [18], we adopted the correlation network proposed in [19], which models the influence of one transmission line to the others in power networks. We then developed a greedy algorithm to build an infection spreading tree starting from a specific node and to assign infection time to each node in order to minimize the cost. We developed two ranking algorithms, cost-based ranking (CR) and



(a) IEEE 300 bus power system

(b) The Great Britain power network

Figure 16: The empirical CDF of the rank of the first tripped transmission line when the failure time of 40% of tripped transmission lines is observed.

tree-based ranking (TR), based on the greedy algorithm. We evaluated the performance of our algorithm on two power systems: IEEE 300 bus power system and the Great Britain power network. We compared our algorithms with existing algorithms for source localization including rumor centrality (RC) [3], Jordan center (ECCE) [2] and NETSLEUTH [4] with 40% timestamps given. The result is in Figure 16, from which we can see that our algorithms outperform others.

References

[1] K. Zhu and L. Ying, "Information source detection in networks: Possibility and impossibility results," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, San Francisco, CA, 2016.

[2] ——, "A robust information source estimator with sparse observations," *Computational Social Networks*, vol. 1, no. 1, p. 3, 2014.

[3] D. Shah and T. Zaman, "Rumors in a network: Who's the culprit?" *IEEE Trans. Inf. Theory*, vol. 57, pp. 5163–5181, Aug. 2011.

[4] B. A. Prakash, J. Vreeken, and C. Faloutsos, "Spotting culprits in epidemics: How many and which ones?" in *IEEE Int. Conf. Data Mining (ICDM)*, Brussels, Belgium, 2012, pp. 11–20.

[5] K. Zhu, Z. Chen, and L. Ying, "CatchEm All: Locating multiple diffusion sources in networks with partial observations," in *AAAI Conference on Artificial Intelligence*, 2017.

[6] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[7] Z. Chen, K. Zhu, and L. Ying, "Detecting multiple information sources in networks under the SIR model," in *Proc. IEEE Conf. Information Sciences and Systems (CISS)*, Princeton, NJ, 2014.

[8] ——, "Detecting multiple information sources in networks under the sir model," *IEEE Transactions on Network Science and Engineering*, vol. 3, no. 1, pp. 17–31, 2016.

[9] K. Zhu and L. Ying, "Information source detection in the SIR model: A sample path based approach," in *Proc. Information Theory and Applications Workshop (ITA)*, Feb. 2013.

[10] ——, "Information source detection in the sir model: A sample-path-based approach," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 408–421, 2016.

[11] ——, "A robust information source estimator with sparse observations," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, Toronto, Canada, April-May 2014.

[12] "Internet autonomous systems network [online]," 2001, available: http://snap.stanford.edu/data/oregon1.html.

[13] "Western states power grid of the united states [online]," 1998, available: http://www-personal.umich.edu/~mejn/netdata/.

[14] A. Y. Lokhov, M. Mezard, H. Ohta, and L. Zdeborova, "Inferring the origin of an epidemy with dynamic message-passing algorithm," *arXiv preprint arXiv:1303.5315*, 2013.

[15] K. Zhu, Z. Chen, and L. Ying, "Locating the contagion source in networks with partial timestamps," *Data Mining and Knowledge Discovery*, vol. 30, no. 5, pp. 1217–1248, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10618-015-0435-9

[16] P. C. Pinto, P. Thiran, and M. Vetterli, "Locating the source of diffusion in large-scale networks," *Phys. Rev. Lett.*, vol. 109, no. 6, p. 068702, 2012.

[17] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos, "Rise and fall patterns of information diffusion: model and implications," in *Proc. Ann. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)*, Beijing, China, 2012, pp. 6–14.

[18] Z. Chen, K. Zhu, and L. Ying, "Root cause localization on power networks," in *IEEE Int. Conf. Digital Signal Processing (DSP)*, Singapor, 2015.

[19] X. Zhang, F. Liu, R. Yao, X. Zhang, S. Mei, Z. Zhang, and X. Li, "Identification of key transmission lines in power grid using modified K-core decomposition," in *Int. Conf. Electric Power and Energy Conversion Systems*, 2013, pp. 1–6.